# Computational Semantics

Assignment 5: Tableaux and Model Checking
Assignment 6: Resolution and Theorem Proving

## Assignment 4.2.1

*$\neg\neg p \to p$*

| | | |
|---|---|---|
| 1. $F(\neg\neg p \to p)$ | ✓ | |
| 2. $T(\neg\neg p)$ | 1, $F\to$ | ✓ |
| 3. $F(p)$ | 1, $F\to$ | |
| 4. $F(\neg p)$ | 2, $T\neg$ | ✓ |
| 5. $T(p)$ | 4, $F\neg$ | |

Unable to falsify, holds in all cases and thus a validity.

*$((p \to q) \to p) \to p$*

| | | |
|---|---|---|
| 1. $F(((p \to q) \to p) \to p)$ | ✓ | |
| 2. $T((p \to q) \to p)$ | 1, $F\to$ | ✓ |
| 3. $F(p)$ | 1, $F\to$ | ✓ |

| | | | | |
|---|---|---|---|---|
| 4. $F(p \to q)$ | 2, $T\to$ ✓ | | 5. $T(p)$ | 2, $T\to$ |
| 6. $T(p)$ | 4, $F\to$ | | | |
| 7. $F(q)$ | 4, $F\to$ | | | |

Unable to falsify because both branches require p to be true, while p had to be false even before the branching.

*$(\neg p \to \neg q) \to (q \to p)$*

| | | |
|---|---|---|
| 1. $F((\neg p \to \neg q) \to (q \to p))$ | ✓ | |
| 2. $T(\neg p \to \neg q)$ | 1, $F\to$ | ✓ |
| 3. $F(q \to p)$ | 1, $F\to$ | ✓ |
| 4. $T(q)$ | 3, $F\to$ | |
| 5. $F(p)$ | 3, $F\to$ | |

| | | | |
|---|---|---|---|
| 6. $F(\neg p)$ | 2, $T\to$ ✓ | 7. $T(\neg q)$ | 2, $T\to$ ✓ |
| 8. $T(p)$ | 6, $F\neg$ | 9. $F(q)$ | 7, $T\neg$ |

Unable to falsify. Before the branching it has been established that q has to be true and p has to be false. In the first branch p has to be true and in the second branch q has to be false, which contradicts what has been said before and thus the model is valid.

*$p \to (p \wedge (q \vee p))$*

| | | |
|---|---|---|
| 1. $F(p \to (p \wedge (q \vee p)))$ | ✓ | |
| 2. $T(p)$ | 1, $F\to$ | |
| 3. $F(p \wedge (q \vee p))$ | 1, $F\to$ | ✓ |

| | | | |
|---|---|---|---|
| 4. $F(p)$ | 3, $F\wedge$ | 5. $F(q \vee p)$ | 3, $F\wedge$ ✓ |
| | | 6. $F(q)$ | 5, $F\vee$ |
| | | 7. $F(p)$ | 5, $F\vee$ |

Unable to falsify because before the branching p had to be true and in both branches p has to be false.

*$(p\vee(q\wedge r))\rightarrow((p\vee q)\wedge(p\vee r))$*

| | | |
|---|---|---|
| 1. $F((p\vee(q\wedge r))\rightarrow((p\vee q)\wedge(p\vee r)))$ | | ✓ |
| 2. $T(p\vee(q\wedge r))$ | 1, $F\rightarrow$ | ✓ |
| 3. $F((p\vee q)\wedge(p\vee r))$ | 1, $F\rightarrow$ | ✓ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4. $T(p)$ | 2, $T\vee$ | | | 5. $T(q\wedge r)$ | 2, $T\vee$ ✓ | | |
| | | | | 6. $T(q)$ | 5, $T\wedge$ | | |
| | | | | 7. $T(r)$ | 5, $T\wedge$ | | |
| 8. $F(p\vee q)$ | 3, $F\wedge$ ✓ | 9. $F(p\vee r)$ | 3, $F\wedge$ ✓ | 10. $F(p\vee q)$ | 3, $F\wedge$ ✓ | 11. $F(p\vee r)$ | 3, $F\wedge$ ✓ |
| 12. $F(p)$ | 8, $F\vee$ | 13. $F(p)$ | 9, $F\vee$ | 14. $F(p)$ | 10, $F\vee$ | 15. $F(p)$ | 11, $F\vee$ |
| 16. $F(q)$ | 8, $F\vee$ | 17. $F(r)$ | 9, $F\vee$ | 18. $F(q)$ | 14, $F\vee$ | 19. $F(r)$ | 11, $F\vee$ |

Unable to falsify because:
In branch one, step 4 dictates p has to be true while in step 12 p has to be false
In branch two, step 4 dictates p has to be true while in step 13 p has to be false
In branch three, step 6 dictates q has to be true while in step 18 q has to be false
In branch four, step 7 dictates r has to be true while in step 19 r has to be false

# Assignment 4.4.1

*$\neg\neg p\rightarrow p$*
Negate:
1. $\neg(\neg\neg p\rightarrow p)$
Convert tot NNF:
2. $\neg(p\rightarrow p)$ (Remove double negations)
3. $p\wedge\neg p$ (Eliminate implications)
Convert to CNF:
4. [[p],[¬p]]

*$((p\rightarrow q)\rightarrow p)\rightarrow p$*
Negate:
1. $\neg(((p\rightarrow q)\rightarrow p)\rightarrow p)$
Convert to NNF:
2. $((p\rightarrow q)\rightarrow p)\wedge\neg p$ (Eliminate implications)
3. $(\neg(p\rightarrow q)\vee p)\wedge\neg p$ (Eliminate implications)
4. $(p\wedge\neg q\vee p)\wedge\neg p$ (Eliminate implications)
Convert to CNF:
5. $((p\vee p)\wedge(\neg q\vee p))\wedge\neg p$ (Distribution rule)
6. [[p,p],[¬q,p],[¬p]]

*$(\neg p\rightarrow\neg q)\rightarrow(q\rightarrow p)$*
Negate:
1. $\neg((\neg p\rightarrow\neg q)\rightarrow(q\rightarrow p))$
Convert to NNF:
2. $(\neg p\rightarrow\neg q)\wedge\neg(q\rightarrow p)$ (Eliminate implications)
3. $(\neg p\rightarrow\neg q)\wedge(q\wedge\neg p)$ (Eliminate implications)
4. $(\neg\neg p\vee\neg q)\wedge(q\wedge\neg p)$ (Eliminate implications)
5. $(p\vee\neg q)\wedge(q\wedge\neg p)$ (Remove double negations)
Convert to CNF:
6. [[p,¬q],[q],[¬p]]

*$p \rightarrow (p \wedge (q \vee p))$*
Negate:
1. $\neg(p \rightarrow (p \wedge (q \vee p)))$

Convert to NNF:
2. $p \wedge (\neg(p \wedge (q \vee p)))$ (Eliminate implications)
3. $p \wedge (\neg p \vee \neg(q \vee p))$ (Drive negations inwards)
4. $p \wedge (\neg p \vee (\neg q \wedge \neg p))$ (Drive negations inwards)

Convert to CNF:
5. $p \wedge ((\neg p \vee \neg q) \wedge (\neg p \vee \neg p))$ (Distribution rule)
6. [[p],[¬p,¬q],[¬p,¬p]]

*$(p \vee (q \wedge r)) \rightarrow ((p \vee q) \wedge (p \vee r))$*
Negate:
1. $\neg((p \vee (q \wedge r)) \rightarrow ((p \vee q) \wedge (p \vee r)))$

Convert to NNF:
2. $(p \vee (q \wedge r)) \wedge (\neg((p \vee q) \wedge (p \vee r)))$ (Eliminate implications)
3. $(p \vee (q \wedge r)) \wedge (\neg(p \vee q) \vee \neg(p \vee r))$ (Drive negations inwards)
4. $(p \vee (q \wedge r)) \wedge ((\neg p \wedge \neg q) \vee (\neg p \wedge \neg r))$ (Drive negations inwards)

Convert to CNF:
5. $((p \vee q) \wedge (p \vee r)) \wedge ((\neg p \wedge \neg q) \vee (\neg p \wedge \neg r))$ (Distribution rule)
6. $((p \vee q) \wedge (p \vee r)) \wedge ((\neg p \vee (\neg p \wedge \neg r)) \wedge (\neg q \vee (\neg p \wedge \neg r)))$ (Associativity rule)
7. $((p \vee q) \wedge (p \vee r)) \wedge ((\neg p \vee \neg p) \wedge (\neg p \vee \neg r) \wedge (\neg q \vee \neg p) \wedge (\neg q \vee \neg r))$ (Distribution rule)
8. [[p,q],[p,r],[¬p,¬p],[¬p,¬r],[¬q,¬p],[¬q,¬r]]

# Exercise 4.4.5

Starting point is the CNF from previous exercise, if needed a conversion to set CNF is done and finally the resolution rule is used for all clause sets.

*[[p],[¬p]]*
[[]] —> one empty clause

*[[p,p],[¬q,p],[¬p]]*
Conversion to set CNF:
[[p],[¬q,p],[¬p]]
[[],[¬q,p]] —> one empty clause

*[[p,¬q],[q],[¬p]]*
[[],[]] —> two empty clauses

*[[p],[¬p,¬q],[¬p,¬p]]*
Conversion to set CNF:
[[p],[¬p,¬q],[¬p]]
[[],[¬p,¬q]] —> one empty clause

*[[p,q],[p,r],[¬p,¬p],[¬p,¬r],[¬q,¬p],[¬q,¬r]]*
Conversion to set CNF:
[[p,q],[p,r],[¬p],[¬p,¬r],[¬q,¬p],[¬q,¬r]]
[[],[],[¬p],[¬q,¬r]] —> two empty clauses

# Axioms in first-order logic for spatial relations

The spatial relations used in the image models are:

- Near
- Part_of
- Supports
- Touch

$\forall x \forall y(\text{SUPPORTS}(x,y) \longrightarrow \text{TOUCH}(x,y))$

$\forall x \forall y(\text{TOUCH}(x,y) \longrightarrow \text{TOUCH}(y,x))$

$\forall x \forall y(\text{NEAR}(x,y) \longrightarrow \text{NEAR}(y,x))$

$\exists x \exists y(\text{TOUCH}(x,y) \longrightarrow \text{SUPPORTS}(x,y))$

$\forall x \forall y(\text{NEAR}(x,y) \longrightarrow \neg(\text{PART\_OF}(x,y) \vee \text{TOUCH}(x,y))$

$\forall x \forall y(\text{PART\_OF}(x,y) \longrightarrow \neg(\text{SUPPORTS}(x,y) \vee \text{NEAR}(x,y) \vee \text{TOUCH}(x,y)))$

$\forall x \forall y \forall z((\text{PART\_OF}(x,y) \wedge \text{PART\_OF}(y,z)) \to \text{PART\_OF}(x,z))$